THE UNIVERSITY OF ALBERTA

THE NUMERICAL SOLUTION OF STIFF DIFFERENTIAL EQUATIONS

by

© SURENDER KUMAR

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

FALL, 1972

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled THE NUMERICAL SOLUTION OF STIFF DIFFERENTIAL EQUATIONS submitted by SURENDER KUMAR in partial fulfillment of the requirements for the degree of Master of Science.

Date July 21, 1972

ABSTRACT

A stiff differential equation is characterized by the fact that the real parts of some of the eigenvalues of the differential equation system are negative and are very much larger in magnitude than the real parts of the other eigenvalues.

Classical difference methods restrict the stepsize due to stability conditions. Many authors have designed new methods, in which a large stepsize can be taken. We have surveyed some of the important methods designed for stiff systems.

Also, we have proposed an improvement on the Liniger and Willoughby method, by developing a fourth order exponentially fitted multistep method. The computational results show that for linear systems with constant coefficients, the proposed method's performance with a stepsize 2h is the same as the performance of the Liniger and Willoughby method with a stepsize h. Even though the proposed method involves more function evaluations than the Liniger and Willoughby method, we conclude that it is more efficient for linear systems. For non-linear systems, the proposed method should only be used, if higher accuracy at the cost of more function evaluations is desired.

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to Dr. L.W. Jackson for his criticism and guidance throughout the duration of this research. Dr. Jackson's suggestions and help really made this thesis possible. I would also like to thank Dr. S. Cabay, for reading and checking the earlier draft of the thesis.

Financial assistance in terms of teaching and research assistantships is gratefully acknowledged to the Department of Computing Science. Finally, my thanks to Mrs. M. Yiu for typing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In physical problems such as those occurring in the study of chemical kinetics, nuclear reactions and control systems, we often encounter "stiff" differential equations, which are difficult to solve even by numerical methods. These stiff equations [5] may be characterized by the fact that the real parts of some of the eigenvalues of the differential equation system are negative and are very much larger in magnitude than the real parts of the other eigenvalues.

Classical difference methods for solving stiff systems are inefficient. Stability requirements which depend on the eigenvalues of the system, force a choice of stepsize, which is too small. Recently however, many authors have introduced new methods [2] designed especially for solving stiff systems. It is the primary objective of this thesis to study these methods and to suggest improvements for the Liniger and Willoughby method [19].

Chapter I is intended to give a general introduction to the problem of stiffness in ordinary differential equations. In Section 1.1, classical single step difference methods are defined and their shortcomings for

solving stiff systems are illustrated. In Section 1.2, multistep methods are defined and some stability definitions to be used in later chapters are given.

Various modified techniques such as Lawson's Runge-Kutta method and the Liniger and Willoughby method are discussed in Chapter II. Gear's method and our proposed fourth order method are described in Chapter III. Finally, in the last chapter, results of test problems using both the Liniger and Willoughby method and our proposed method are given. It is shown that for linear systems with constant coefficients, the proposed method's performance with a stepsize 2h is same as the performance of Liniger and Willoughby third order method with a stepsize h. The principal advantage of the proposed method is that of higher accuracy. However, the cost of function evaluations is more than the Liniger and Willoughby method. The tradeoff for one non-linear system is discussed in the last chapter.

1.1 Single Step Methods for Stiff Systems

Definition

Any method for solving a differential equation

$$\dot{y} = f(x,y) \tag{1.1.1}$$

in which the approximation y_{n+1} to the solution at the point x_{n+1} can be calculated from only x_n, y_n and $h = x_{n+1} - x_n$ is called a single step method.

By a stable method, we mean one for which the error $\varepsilon_n = y_n - y(x_n)$ remains bounded as $n \to \infty$. To see the problem that arises, when using single step methods for solving stiff equations, consider the equation

$$\dot{y}(x) = \lambda(y - F(x)) + \dot{F}(x), \quad \lambda < 0 \tag{1.1.2}$$

with the solution

$$y(x) = c.\exp(\lambda x) + F(x) \tag{1.1.3}$$

where c is a parameter. Assuming that F(x) is not itself exponential, we see that the term $c.\exp(\lambda x)$ is significant for small x, but is insignificant for large x. Nevertheless, the parameter λ appearing in $c.\exp(\lambda x)$, because of stability considerations, restricts the stepsize both for small and large x.

For example, on applying Euler's method $y_{n+1} = y_n + hf_n$ to (1.1.2) with $F(x) = 0$, we obtain

$$y_{n+1} = y_n(1 + \lambda h) \ .$$

Since $y(x) \to 0$ as $x \to \infty$, Euler's method is stable if and only if $0<|h\lambda|<2$. The behaviour of y_n in the case of instability i.e., $|h\lambda|>2$ is illustrated in Fig.1† for general $F(x)$. The stability restriction on h, must be maintained for all x, even though the term in (1.1.3) containing λ becomes negligible for large x. The inefficiency of Euler's method for solving (1.1.2), arises from the fact that for large x, stability requirements force a choice of stepsize, which is too small.

This difficulty persists with higher order single step methods, such as the Runge-Kutta fourth order method. On applying the R-K 4 method

$$y_{n+1} = y_n + (K_0 + 2K_1 + 2K_2 + K_3)/6$$

where

$$K_0 = hf(y_n) \ , \quad K_1 = hf(y_n + K_0/2) \ ,$$

$$K_2 = hf(y_n + K_1/2) \quad \text{and} \quad K_3 = hf(y_n + K_2)$$

to the equation $\dot{y} = \lambda y$, we obtain

---

† All figures are given at the end of the section.

$$y_{n+1} = P(h\lambda)y_n$$

where

$$P(h\lambda) = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{6} + \frac{(h\lambda)^4}{24} \quad .$$

For stability, we must have

$$|P(h\lambda)| < 1$$

which gives the bound $0<|h\lambda|<2.78$, when λ is real. The stability region for the complex λ is shown in Fig. 2.

We now extend the above arguments to linear systems of differential equations

(1.1.4) $$\dot{y} = Ay$$

with the initial conditions $y(0) = y_0$, where A is an $n \times n$ matrix with n distinct eigenvalues. If the eigenvalues of A are in the negative half plane (Real $\lambda < 0$), the solution of the system is said to be inherently stable.

For $h > 0$ and $x_n = nh$, $n = 0,1,2,\ldots,N$, the solution of (1.1.4) is

(1.1.5) $$y(x_n) = [\exp(hA)]^n y_0 \quad .$$

Since the eigenvalues of A are distinct, there exists a matrix P such that

(1.1.6) $$A = P \Lambda P^{-1}$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. The solution of (1.1.4) can then be written as

$$y(x_n) = P[\exp(h\Lambda)]^n P^{-1} y_0 . \tag{1.1.7}$$

If either Euler's method or the R-K 4 method is used, the approximate solution is

$$y_n = [M(hA)]^n y_0 \tag{1.1.8}$$

where

$$M(hA) = \begin{cases} I + hA \\ \\ I + hA + \dfrac{h^2A^2}{2} + \dfrac{h^3A^3}{6} + \dfrac{h^4A^4}{24} \end{cases}$$

respectively. With A given by (1.1.6), equation (1.1.8) becomes

$$y_n = P[M(h\Lambda)]^n P^{-1} y_0 . \tag{1.1.9}$$

Since Λ is a diagonal matrix,

$$M(h\Lambda)_{ii} = \begin{cases} 1 + h\lambda_i \\ \\ 1 + h\lambda_i + \dfrac{h^2\lambda_i^2}{2} + \dfrac{h^3\lambda_i^3}{6} + \dfrac{h^4\lambda_i^4}{24} \end{cases}$$

and we see that $M(h\Lambda)_{ii}$ is a truncated Taylor series expansion of $\exp(h\Lambda)$.

Comparing equations (1.1.7) and (1.1.9), it is clear that Euler's method and the R-K 4 method respectively

are unstable, if and only if for some i

$$|1 + h\lambda_i| > 1$$

and

$$\left|1 + h\lambda_i + \frac{h^2\lambda_i^2}{2} + \frac{h^3\lambda_i^3}{6} + \frac{h^4\lambda_i^4}{24}\right| > 1 .$$

To make this remark clear, we consider the following examples from two space:

(1.1.10) $\dot{y} = Ay \qquad y(0) = [0.9, 1.1]^t$

(1.1.11) $\dot{z} = Dz \qquad z(0) = [1, 1]^t$

where

$$A = \begin{pmatrix} -500.5 & 499.5 \\ 499.5 & -500.5 \end{pmatrix} \quad \text{and} \quad D = \text{diag}(-1, -1)$$

The solution of (1.1.10) is

$$y_1(x) = \exp(-x) - 0.1 \exp(-1000\ x)$$

$$y_2(x) = \exp(-x) + 0.1 \exp(-1000\ x)$$

and that of (1.1.11) is

$$z_1(x) = \exp(-x)$$

$$z_2(x) = \exp(-x) \quad .$$

Rewriting $y_1(x)$ and $y_2(x)$, we have

$$y_1(x) = z_1(x) - 0.1 \exp(-1000\, x)$$

$$y_2(x) = z_2(x) + 0.1 \exp(-1000\, x) \ .$$

The graph of the solutions is given in Fig. 3.

Observe that, except for those points in some neighbourhood of the origin, the solutions $y(x)$ and $z(x)$ are practically identical. Indeed on most computers, single precision computations would completely ignore the transient term $\exp(-1000\, x)$ for $x > 0.02$ ($\exp(-1000\, x) \approx 2.06 \times 10^{-9}$, when $x = 0.02$). We should therefore expect that any particular method, when applied to (1.1.10) or (1.1.11), should be equally effective at least for $x > 0.02$.

That this is not the case with classical difference methods, is a consequence of the great disparity in the eigenvalues of the two systems ($\lambda_1 = \lambda_2 = -1$ in (1.1.11), while $\lambda_1 = -1$, $\lambda_2 = -1000$ in (1.1.10)). Thus, although the transient term $\exp(-1000\, x)$ is negligible for $x > 0.02$, it is precisely this term, obtained from the eigenvalue $\lambda_2 = -1000$, which determines the maximum permissible stepsize for a stable difference scheme. For example, if Euler's method is applied to (1.1.11), the stability condition is

$$\max|1 + \lambda_i h| = |1 - h| < 1 \ , \qquad (1.1.12)$$

which is satisfied if $|h| < 2$, but if it is applied to (1.1.10), the stability condition is

$$(1.1.13) \qquad \max|1 + \lambda_i h| = |1 - 1000\ h| < 1\ ,$$

which is satisfied if $|h| < 2 \times 10^{-3}$. When solving (1.1.10) by Euler's method, restriction (1.1.13) on the stepsize may indeed be reasonable, since in this interval, the eigenvalue $\lambda_2 = -1000$ is certainly significant. Outside this interval however, equation (1.1.10) is essentially the same as (1.1.11) and the more reasonable restriction (1.1.12) on the stepsize should be imposed. In fact, methods designed for stiff systems, ignore the transient terms, as the computation proceeds. This fact is what distinguishes them from the classical methods.

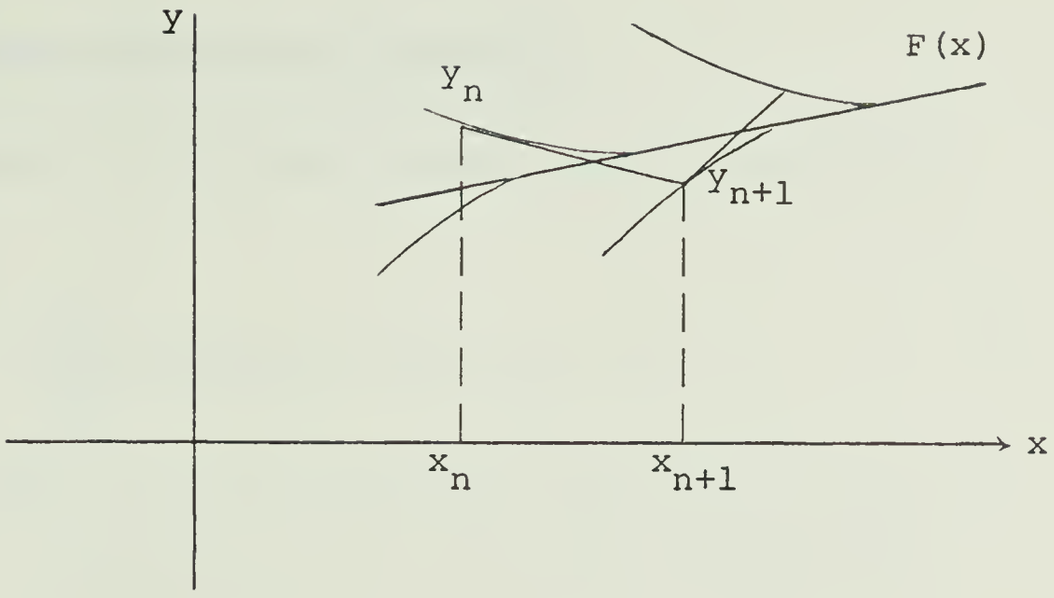

Fig.1 Instability of Euler's Method

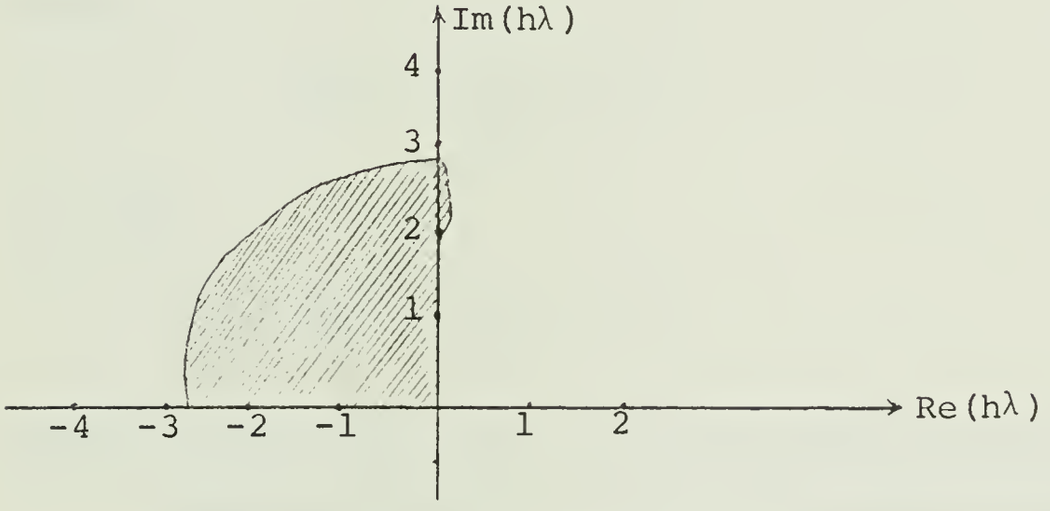

Fig.2 Upper Stability Region for R-K 4 Method

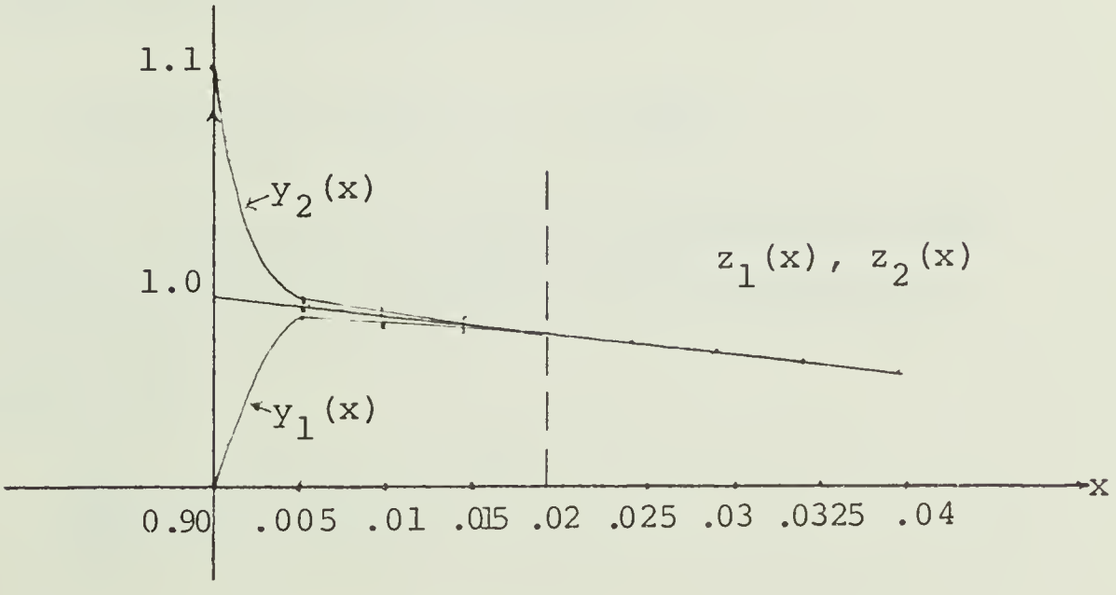

Fig.3 The Behaviour of y and z

1.2 Multistep Methods and Stability

Define the multistep method for (1.1.1) by the equation

$$\sum_{j=0}^{k} (\alpha_j y_{n-j} + h\beta_j f_{n-j}) = 0 \tag{1.2.1}$$

where α_k and β_k are real constants with $|\alpha_k| + |\beta_k| \neq 0$ and $\alpha_0 \neq 0$. Equation (1.2.1) is commonly expressed as

$$\rho(E)y_{n-k} + h\sigma(E)f_{n-k} = 0$$

where

$$\rho(\mu) = \sum_{j=0}^{k} \alpha_j \mu^{k-j} \quad ,$$

$$\sigma(\mu) = \sum_{j=0}^{k} \beta_j \mu^{k-j} \tag{1.2.2}$$

and E is the usual shift operator defined by $Ey_n = y_{n+1}$.

Applying method (1.2.1) to the single equation $\dot{y} = \lambda y$, yields the numerical solution

$$y_n = g_1\mu_1^n + g_2\mu_2^n + \ldots + g_k\mu_k^n$$

where $g_1, g_2, \ldots, g_k$ are constants depending on the initial values and the μ_k satisfy the characteristic equation given by

$$\sum_{j=0}^{k} (\alpha_j + h\lambda\beta_j) \mu^{k-j} = 0 \; . \tag{1.2.3}$$

If μ_1 is the principal root, then for small h

$$\mu_1 = \exp(h\lambda) + 0(h^{p+1})$$

where p is the order of the method. If $\lambda < 0$, the numerical solutions should tend to zero as $n \to \infty$. Thus, if a method is applied to any inherently stable system, the principal root of (1.2.3) should be less than one. Dahlquist [6] has defined this property to be A-stability.

Definition

A k-step method with fixed stepsize $h > 0$ is A-stable, if the numerical solution of any inherently stable differential equation tends to zero as $n \to \infty$.

If $|\mu_1| > |\mu_j|$, $j=2,3,\ldots,k$, the solutions determined by the k-1 extraneous roots $\mu_2, \mu_3, \ldots, \mu_k$ are small compared with the principal solution $g_1\mu_1^n$. However, if $|\mu_1| < |\mu_j|$ for any j, then the method is unstable. We note that for $\lambda < 0$, $y(x) \to 0$ as $x \to \infty$ and therefore the roots μ_k must be in the unit circle.

Definition : Gear [12]

A multistep method is stable if the roots μ_j of the polynomial $\rho(\mu)$ are such that $|\mu_j| \leq 1$ and those roots which satisfy $|\mu_j| = 1$ are simple.

Definition : Gear [12]

A multistep method is absolutely stable for those values of $h\lambda$, where roots of (1.2.3) are ≤ 1 in absolute value.

Dahlquist [6] has shown that the order p of an A-stable multistep method cannot exceed 2 and that the smallest truncation error of those methods of order 2, is obtained by the trapezoidal rule (see Fig.4 for the A-stability domain). Thus, if multistep methods of order $p > 2$ are to be used, the constraints imposed by A-stability must be relaxed.

Several authors have given stability definitions that relax the requirements of A-stability. These definitions and methods associated with them are surveyed in Bjurel et al. [2]. In this thesis, in addition to A-stability, we shall use the concept of stiff stability given by Gear [11].

Definition

A method is stiffly stable if it is absolutely stable in the region R_1 ($Re(h\lambda) \leq \delta_1$) (see Fig.5) and is accurate in the region R_2 ($\delta_1 < Re(h\lambda) < \delta_2$, $|Im(h\lambda)| < \gamma$).

We are only interested in stability for the decaying solutions of an inherently stable system, since we choose h such that $\exp(h\lambda)$ is very small. We are concerned both with accuracy and with stability for the other roots. If $h\lambda = x + iy$, Gear shows that the maximum stable and accurate region is a rectangle bounded by $\delta_1 \leq x \leq \delta_2$ and $-\gamma \leq y \leq \gamma$, where $\exp(\delta_1)$ is the maximum amount of

growth allowed in one step and γ is chosen so that the method is accurate for the imaginary components.

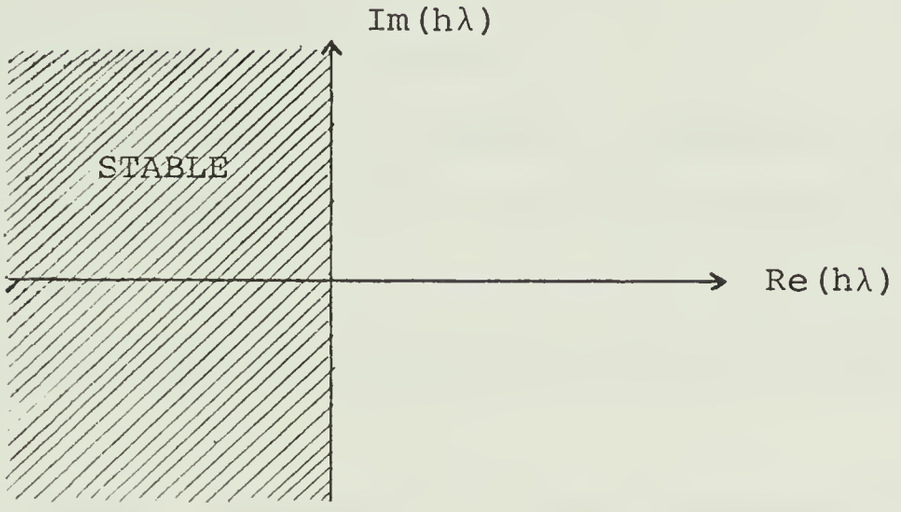

Fig.4 A-stability Domain

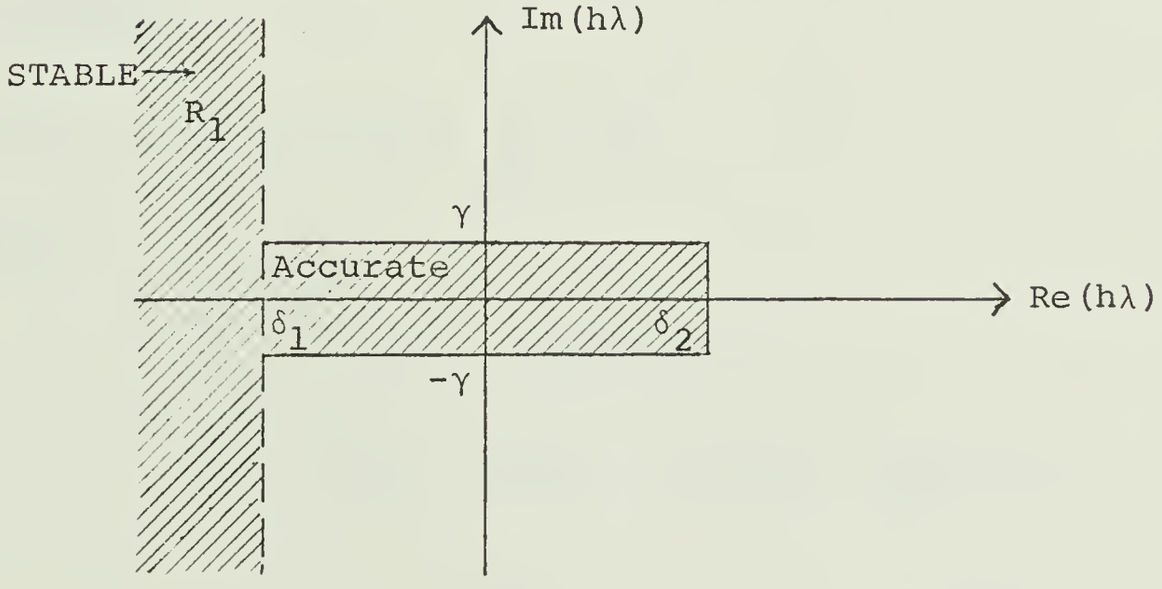

Fig.5 Stiffly Stable Domain

CHAPTER II

SINGLE STEP METHODS

2.1 Modified Runge-Kutta Methods

For the explicit R-K 4 method of section (1.1), stability requirements for (1.1.1) force a choice of stepsize which is too small. To increase stability, several authors have proposed modifications to the R-K methods.

For example, Rosenbrock [22] has formulated the semi-implicit R-K method for

$$\dot{y} = f(y) \tag{2.1.1}$$

as follows:

$$y_{n+1} = y_n + \sum_{j=1}^{p} w_j K_j \tag{2.1.2}$$

where

$$K_1 = hf(y_n) + ha_1 A(y_n) K_1 ,$$

$$K_2 = hf(y_n + b_{21} K_1) + ha_2 A(y_n + c_{21} K_1) K_2 ,$$

. . . .

$$K_p = hf(y_n + \sum_{j=1}^{p-1} b_{pj} K_j) + ha_p A(y_n + \sum_{j=1}^{p-1} c_{pj} K_j) K_p$$

and $A(y) = \partial f(y)/\partial y$.

In his paper, Rosenbrock suggests one possible choice of the coefficients a_p, b_{pj} and c_{pj}, which are summarized in row 1 of Table 1. Calahan [4] suggests an alternative choice of coefficients, which are summarized in row 2 of Table 1. Lapidus et al. [15] report that the method with Calahan's coefficients is computationally faster than most of the single step methods for moderately stiff systems. However, for a highly stiff system, the method is computationally slower than other available methods, such as the Liniger and Willoughby method considered in section 2.2. Also, Lawson's method [16] (see pp. 17) should be better for linear systems than Rosenbrock's method with Calahan's parameters.

Table 1

The Coefficients of Rosenbrock's Method

Method	$a_1=a_2$	b_{21}	c_{21}	w_1	w_2	Stage	Order
Rosenbrock	$1-1/\sqrt{2}$	$(\sqrt{2}-1)/2$	0	0	1	2	2
Calahan	0.78868	-1.15470	0	0.75	0.25	2	3

To provide the reader with an intuitive notion of the difference between Rosenbrock's method and the classical R-K methods, we apply Rosenbrock's method to $\dot{y} = \lambda y$.

Equation (2.1.2) now becomes

$$y_{n+1} = R(Q)y_n$$

where $Q = h\lambda$. Corresponding to Rosenbrock's and to Calahan's choice of coefficients, we have

$$R(Q) = \begin{cases} \dfrac{1 + (\sqrt{2} - 1)Q}{1 + (\sqrt{2}-2)Q + (1.5 - \sqrt{2})Q^2} \\ \\ \dfrac{1 - 0.578Q - 0.456Q^2}{1 - 1.578Q + 0.622Q^2} \end{cases}$$

respectively.

We observe therefore that while classical R-K methods produce polynomial approximations to exp(Q) (c.f., section 1.1), Rosenbrock's method produces rational approximations to exp(Q). Thus, since we might expect that functions with transient terms can be more easily approximated by rational functions than by polynomials, Rosenbrock's method should be better for solving stiff systems than the classical R-K methods.

Another modification due to Lawson of the classical R-K method can be developed by using the following transformation

$$z(x) = \exp(-xA)y(x) \tag{2.1.3}$$

where A is a real matrix. The motivation for using this transformation is to separate a stiff system into a

linear part and a transient part. Equation (1.1.1) after transformation becomes

$$\dot{z}(x) = \exp(-xA)(f(x,\exp(xA)z(x)) - A\exp(xA)z(x)) \equiv g(x,z) \quad . \tag{2.1.4}$$

The Jacobian of (2.1.4) is

$$\partial g/\partial z = \exp(-xA)(\partial f/\partial y - A)\exp(xA) \quad .$$

Thus the eigenvalues of $\partial g/\partial z$ are those of $\partial f/\partial y - A$. Applying the general R-K method to (2.1.4) yields

$$z_{n+1} = z_n + \sum_{j=1}^{p} w_j K_j \qquad j=2,3,\ldots,p$$

where

$$K_1 = \exp(-x_nA)(f(x_n,\exp(x_nA)z_n) - A\exp(x_nA)z_n) \quad ,$$

$$p_j = \exp((x_n + c_jh)A)(z_n + h\sum_{i=1}^{j-1} a_{ij}K_i)$$

and

$$K_j = \exp(-(x_n + c_jh)A)(f(x_n + c_jh,p_j) - Ap_j) \quad .$$

On transforming to the y-coordinate system, we obtain

$$y_{n+1} = \exp(hA)y_n + h\sum_{j=1}^{p} w_j\exp((1-c_j)hA)K_j^* \tag{2.1.5}$$

where

$$K_1^* = f(x_n,y_n) - Ay_n \quad ,$$

$$p_j^* = \exp(c_jhA)y_n + h\sum_{i=1}^{j-1} a_{ij}\exp((c_j-c_i)hA)K_i^*$$

and

$$K_j^* = f(x_n + c_j h, p_j^*) - Ap_j^*$$

for $0 = c_1 \leq c_2 \leq \ldots \leq c_p \leq 1$.

Lawson proves that the method is A-stable. System (2.1.5) is used computationally for integration and Diagonal Padé approximants are used for approximating $\exp(c_j hA)$. The Jacobian matrix $\partial f/\partial y$, which is to be re-evaluated frequently, is used for the matrix A. The main advantage of the method is the ability of transforming a stiff system into a simpler non-stiff system, which is easy to integrate. The advantage, however, must be weighed against the disadvantage of evaluating the Jacobian.

2.2 Liniger and Willoughby Method

Liniger and Willoughby [19] developed integration formulas based upon the following identities

$$(2.2.1a) \qquad y(x+h)-y(x)-h((1-\mu)f(x+h)+\mu f(x)) \equiv e_1(x) ,$$

$$(2.2.1b) \qquad y(x+h)-y(x)-h((1+a)f(x+h)+(1-a)f(x))/2 + h^2((b+a)\dot{f}(x+h)-(b-a)\dot{f}(x))/4 \equiv e_2(x)$$

and

$$(2.2.1c) \qquad y(x+h)-y(x)-h((1+c)f(x+h)+(1-c)f(x))/2 + h^2((1+3c)\dot{f}(x+h)-(1-3c)\dot{f}(x))/12 \equiv e_3(x)$$

where

$$e_1(x) = -h^2 \int_0^1 (\theta-\mu)\dot{f}(x+\theta h)d\theta ,$$

$$e_2(x) = (h^3/4) \int_0^1 (2\theta^2-2(1-a)\theta + (b-a))\ddot{f}(x+\theta h)d\theta$$

and

$$e_3(x) = -(h^4/12) \int_0^1 \theta(2\theta^2-3(1-c)\theta + (1-3c))\dddot{f}(x+h)d\theta .$$

Here μ, a, b and c are real parameters. These formulas lead naturally to the following finite difference schemes (which we label $F^{(1)}$, $F^{(2)}$ and $F^{(3)}$ for convenience):

$$(2.2.2a) \qquad F^{(1)}: y_{n+1} = y_n + h((1-\mu)f_{n+1} + \mu f_n) ,$$

(2.2.2b) $$F^{(2)}: y_{n+1} = y_n + h((1+a)f_{n+1} + (1-a)f_n)/2 - h^2((b+a)\dot{f}_{n+1} - (b-a)\dot{f}_n)/4$$

and

(2.2.2c) $$F^{(3)}: y_{n+1} = y_n + h((1+c)f_{n+1} + (1-c)f_n)/2 - h^2((1+3c)\dot{f}_{n+1} - (1-3c)\dot{f}_n)/12 .$$

We note that formula $F^{(i)}$ is of order i and $F^{(3)}$ is a special case of $F^{(2)}$ with b = 1/3 and c = a.

In order to find a rationale for choosing the parameters in the above formulas, we follow the ideas of Liniger and Willoughby and apply the formulas to the equation $\dot{y} = \lambda y$; $Re(\lambda) < 0$. The true solution is given by

(2.2.3) $$y_{n+1} = \exp(Q)y_n$$

where $Q = h\lambda$.

Corresponding to equations (2.2.2), the results obtained in each case are of the form

(2.2.4) $$y_{n+1} = r^{(i)}(Q)y_n$$

where

(2.2.5a) $$r^{(1)}(Q) = (1 + \mu Q)/(1 - (1-\mu)Q) ,$$

(2.2.5b) $$r^{(2)}(Q) = (4 + 2(1-a)Q + (b-a)Q^2)/(4 - 2(1+a)Q + (b+a)Q^2)$$

and

$$(2.2.5c) \qquad r^{(3)}(Q) = (12 + 6(1-c)Q + (1-3c)Q^2)/$$
$$(12 - 6(1+c)Q + (1+3c)Q^2)$$

respectively.

For A-stability, we must have $|r^{(i)}(Q)| \leq 1$, which gives the following conditions for $F^{(i)}$:

$$(2.2.6a) \qquad F^{(1)}: \quad \mu \leq 0.5 \quad ,$$

$$(2.2.6b) \qquad F^{(2)}: \quad a \geq 0 \quad , \quad b \geq 0$$

and

$$(2.2.6c) \qquad F^{(3)}: \quad c \geq 0 \quad .$$

Let $\varepsilon^{(i)}(Q) = r^{(i)}(Q) - \exp(Q)$ be the error for method $F^{(i)}$.

Definition

If the free parameters are chosen such that the formulas $F^{(i)}$ are made exact in a discrete sense (i.e., $\varepsilon^{(i)}(Q_0) = 0$ for a particular $Q_0 = h\lambda_0$), then the formulas $F^{(i)}$ are said to be exponentially fitted at $Q = Q_0$.

By definition, we require the following relations to hold for particular Q_0's:

$$(2.2.7a) \qquad \varepsilon^{(1)}(Q_0) = 0 \quad ,$$

$$(2.2.7b) \qquad \varepsilon^{(2)}(Q_0) = \varepsilon^{(2)}(Q_1) = 0$$

and

$$\varepsilon^{(3)}(Q_0) = 0 \qquad (2.2.7c)$$

where Q_0 and Q_1 are obtained from two different eigenvalues λ_0 and λ_1.

The solution of each equation in (2.2.7) gives the following values for the parameters:

$$\mu = -1/Q - 1/(\exp(-Q) - 1) \ , \qquad (2.2.8a)$$

$$a = 2(v(Q_1) - v(Q_0))/(Q_1 v(Q_0) - Q_0 v(Q_1)) \ , \qquad (2.2.8b)$$

$$b = 2(Q_0 - Q_1)/(Q_0 v(Q_1) - Q_1 v(Q_0)) \qquad (2.2.8c)$$

and

$$c = \frac{(Q^2 + 6Q + 12 - \exp(Q)(Q^2 - 6Q + 12))}{3(\exp(Q)(Q^2 - 2Q) + Q^2 + 2Q)} \qquad (2.2.8d)$$

where

$$v(z) = (z^2(1 - \exp(-z))/(-(2+z) + (2+z)\exp(-z)) \ .$$

We know that conventional methods restrict the stepsize in the early time period. The main advantage of this method is that the same stepsize used in the later time period, is used during the early time period of integration. Another advantage is that the free parameters can be put to zero after the transient terms do not contribute, thus getting higher order methods.

2.3 Dahlquist's Smoothing Technique

Consider the differential equation

(2.3.1) $\dot{y} = \lambda y \ , \quad Re(\lambda) < 0 \ , \quad y(0) = 1$

with the true solution

(2.3.2) $y(x) = \exp(\lambda x) \ .$

The approximate solution of (2.3.1) at $x_n = nh$ obtained using the trapezoidal rule is $y_n = ((1+z)/(1-z))^n$ where $z = h\lambda/2$. Clearly, for $-z$ very large, the solution points oscillate about zero. To damp these oscillations, Dahlquist has introduced the idea of smoothing. Smoothing consists of replacing solution points y_n with weighted averages of solution points in the neighbourhood of y_n. For example, one particular smoothing formula is

(2.3.3) $\hat{y}_n = (y_{n-1} + 2y_n + y_{n+1})/4 \ .$

Dahlquist suggests using this formula one or more times in calculating the solution. To define a typical smoothing algorithm, let $\{n_j\}_{j=0}^{m}$ be any sequence of integers satisfying $n_0 = 0$, $n_j \geq n_{j-1} + 1$ and $n_m < N$, where m is the maximum number of times (2.3.3) is used and y_{n_1}, y_{n_2}, ..., y_{n_m} represents the solution points at which smoothing is desired. The smoothing algorithm for (2.3.1) is given below. The input consists of m, n_1, n_2,...,n_m, N, z and y_0.

1. [READ] READ $m, n_1, \ldots, n_m$, N, z, y_0.

2. [INITIALIZE] $j \leftarrow 1$;

$y \leftarrow y_0$.

3. [TRAPEZOIDAL RULE] $y \leftarrow y((1+z)/(1-z))^{n_j - 1}$.

4. [SMOOTHING] $y \leftarrow (1 + 2(\frac{1+z}{1-z}) + (\frac{1+z}{1-z})^2)\frac{y}{4}$.

5. [INCREMENT] $j \leftarrow j + 1$;

IF $j \leq m$ THEN GOTO 3.

6. [FINISH] $K \leftarrow N - n_m + 1$.

7. [DECREMENT] $K \leftarrow K - 1$;

IF $K = 0$ THEN STOP.

8. [UPDATE] $y \leftarrow y(1+z)/(1-z)$;

GO TO 7.

For the preceding algorithm, one can show (Lindberg [17]) that the computed solution points y_n satisfy

$$(2.3.4) \qquad \hat{y}_n = ((1+z)/(1-z))^n/(1-z^2)^m = y_n/(1-z^2)^m.$$

It is important to notice that the amplitude of the oscillations of $\hat{y}_n$ is smaller than those of y_n, by the factor $(1-z^2)^m$. However, smoothing methods introduce new errors. For the above algorithm, the error ε_1 is given by

(2.3.5a) $$\varepsilon_1(n) = \exp(nh\lambda) - y_n/(1-z^2)^m \ .$$

The error ε_2 of the trapezoidal rule defined by

(2.3.5b) $$\varepsilon_2(n) = \exp(nh\lambda) - y_n$$

and the error ε_1 satisfy

(2.3.6) $$\varepsilon_1(n) = \varepsilon_2(n) + y_n(1 - 1/(1-z^2)^m) \ .$$

From (2.3.6), we see that for $|z| < 1$, the error ε_1 is smaller than ε_2. However, for $|z| > 1$, equation (2.3.6) clearly shows that the amplitude of the error will be growing. We note that as m increases, the amplitude of the oscillations gets smaller. However, Lindberg shows that for the non-stiff components (components associated with small eigenvalues), accuracy is lost as m increases. Since smoothing itself creates new errors, this error is proportional to the error in the trapezoidal rule for linear systems with constant coefficients. For non-linear systems with large initial values, the errors may get amplified.

Lindberg combines the smoothing technique with extrapolation and shows that without extrapolation, smoothing cannot be done with the trapezoidal rule. We note that many other smoothing formulas exist, but only those which introduce errors commensurate with the errors made by the integration formula are acceptable.

CHAPTER III

MULTISTEP METHODS

3.1 Gear's Method

For the system (1.1.1), consider the multistep method

$$(3.1.1) \qquad y_{n+1} = -\sum_{j=1}^{k} \alpha_j y_{n+1-j} - h \sum_{j=0}^{k} \beta_j f_{n+1-j}$$

where $\beta_0 \neq 0$ and $|\alpha_k| + |\beta_k| \neq 0$.

For stability analysis, we are interested in the behaviour of the roots of the polynomial

$$(3.1.2) \qquad \rho(\xi) + h\lambda_j \sigma(\xi) = 0$$

where λ_j are the eigenvalues of the Jacobian matrix $\partial f/\partial y$. The stability domain of (3.1.1) is the set of $h\lambda$ for which the roots of (3.1.2) are inside the unit circle. The boundary of the stability domain is given by the locus of $-\rho(\xi)/\sigma(\xi)$ for $\xi = \exp(i\theta)$, $\theta \in [0,2\pi]$.

Gear [11] observed that as $h\lambda$ changes from 0 to $-\infty$, the roots of (3.1.2) move from the roots of $\rho(\xi)$ to those of $\sigma(\xi)$. The most stable $\sigma(\xi)$ polynomial of degree k is $\sigma(\xi) = \xi^k$. By using the stability domains, Gear has shown that for $k \leq 6$, these methods are stiffly stable. For example, if $k = 6$ then δ_1 satisfies $\delta_1 < -6.1$ and γ

satisfies $\gamma < 0.5$. Jain and Srivastva [12] have shown the existence of stiffly stable methods upto order 11.

Using the corrector formula (3.1.1), we obtain the fixed point iteration

$$(3.1.3) \qquad y_{n+1}^{(m+1)} = -h\ \beta_0\ f(x_{n+1},\ y_{n+1}^{(m)}) - C$$

where

$$C = \sum_{j=1}^{k} [\alpha_j\ y_{n+1-j} + h\beta_j f_{n+1-j}] \ .$$

If the predicted value of y_{n+1} is $y_{n+1}^{(0)}$, this fixed point iteration may not converge for large values of $h\partial f/\partial y$, because the normal convergence condition is $|h\beta_0 \partial f/\partial y| < 1$.

This restriction can be avoided by reformulating the iteration (3.1.3) as

$$(3.1.4a) \qquad T^{(m)} y_{n+1}^{(m+1)} = -h\beta_0 [f(y_{n+1}^{(m)}) - D^{(m)} y_{n+1}^{(m)}] - C$$

where

$$T^{(m)} = [I + h\beta_0 D^{(m)}]$$

and $D^{(m)}$ is a suitably chosen matrix.

To describe how $D^{(m)}$ should be chosen, we show that if $D^{(m)} = \partial f/\partial y$, evaluated at $y_{n+1}^{(m)}$, then (3.1.4a) is a Newton iteration.

Dropping the subscripts, we can write (3.1.3) as

$$y = -h\beta_0 f(y) - C \ .$$

Using Newton's method to find y, we obtain

$$(3.1.4b) \qquad [I + h\beta_0 \partial f^{(m)}/\partial y](y^{(m+1)} - y^{(m)})$$

$$= -(y^{(m)} + h\beta_0 f(y^{(m)}) + C) \ .$$

By adding $y^{(m)} + h\beta_0 D^{(m)} y^{(m)}$ to both sides of (3.1.4b) and rearranging terms, we obtain

$$(3.1.4c) \qquad T^{(m)} y^{(m+1)} + h\beta_0 (y^{(m+1)} (\partial f^{(m)}/\partial y - D^{(m)})$$

$$+ y^{(m)} (D^{(m)} - \partial f^{(m)}/\partial y))$$

$$= -h\beta_0 (f(y^{(m)}) - D^{(m)} y^{(m)}) - C \ .$$

Choosing the matrix $D^{(m)}$ to be the Jacobian matrix $\partial f/\partial y$, evaluated at $y^{(m)}$, yields the required result.

For computational purposes, Gear reformulates equation (3.1.4a) for $m \geq 1$ as follows:

$$(3.1.5) \qquad y^{(m+1)} = y^{(m)} - \beta_0 [T^{(m)}]^{-1} (hf(y^{(m)}) - d^{(m)})$$

where

$$d^{(m)} = hf(y^{(m-1)}) + hD^{(m-1)} (y^{(m)} - y^{(m-1)}) \ .$$

Furthermore, he finds that given $d^{(1)}$, the sequence $\{d^{(m)}\}_{m=1}^{\infty}$ satisfies

$$(3.1.6) \qquad d^{(m+1)} = d^{(m)} + [T^{(m)}]^{-1} (hf(y^{(m)}) - d^{(m)}) \ .$$

In fact, Gear shows that (3.1.5) and (3.1.6) can be used for $m \geq 0$ provided that

$$d^{(0)} = -(y^{(0)} + \sum_{j=1}^{k} (\alpha_j y_{n+1-j} + h\beta_j y_{n+1-j}))/\beta_0 \ .$$

Gear [13] has shown that any general multistep method (3.1.1) can be written as a multivalue method. Thus if (3.1.5) is written in the multivalue form, we obtain

$$\underline{z}_{n+1}^{(m+1)} = \underline{z}_{n+1}^{(m)} + \underline{b}.F(\underline{z}_{n+1}^{(m)}) \ , \tag{3.1.7a}$$

$$\underline{z}_{n+1}^{(0)} = B\underline{z}_n \tag{3.1.7b}$$

and

$$\underline{z}_{n+1} = \underline{z}_{n+1}^{(M)} \tag{3.1.7c}$$

where

$$\underline{z}_n^{(m)} = [y_n^{(m)}, y_{n-1}, y_{n-2}, \cdots, y_{n-k} ,$$

$$d_n^{(m)}, hf_{n-1}, \ldots, hf_{n-k}]^t \ ,$$

$$\underline{b} = [-\beta_o , 0 , \ldots, 0 , 1 , 0 , \ldots 0]^t ,$$

$$F(\underline{z}_{n+1}^{(m)}) = [T^{(m)}]^{-1}(hf(y_{n+1}^{(m)}) - d_{n+1}^{(m)})$$

and the matrix B is given by

$$B = \begin{pmatrix} \alpha_1^* & \cdots & \alpha_k^* & \beta_1^* & \cdots & \beta_k^* \\ 1 & & 0 & & 0 & \\ & \ddots & & & & \\ 0 & & 1\ 0 & & & \\ \hline \eta_1 & \cdots & \eta_k & \phi_1 & \cdots & \phi_k \\ & 0 & & 1 & & \\ & & & & \ddots & \\ & & & 0 & & 1\ 0 \end{pmatrix}$$

The coefficients in B are given by

$$\eta_j = -(\alpha_j^* + \alpha_j)/\beta_0 \quad ,$$

$$\phi_j = -(\beta_j^* + \beta_j)/\beta_0 \quad ,$$

where α_j^*, β_j^* are the coefficients associated with the following predictor formula

$$y_{n+1}^{(0)} = \sum_{j=1}^{k} (\alpha_j^* y_{n+1-j} + h\beta_j^* f_{n+1-j}) \quad .$$

Gear reformulates the multivalue method (3.1.7), using standard techniques (Gear [12]) into a multivalue formulation using vectors $\underline{a}_n$ of scaled derivatives given by

$$\underline{a}_n = [y_n, h\dot{y}_n, h^2\ddot{y}_n/2!, \ldots, h^{k-1}y_n^{(k-1)}/(k-1)!]^t \quad .$$

In the Nordsieck formulation, the multivalue method has the form

(3.1.8a) $$\underline{a}_{n+1}^{(0)} = A\,\underline{a}_n \quad ,$$

(3.1.8b) $$\underline{a}_{n+1}^{(m+1)} = \underline{a}_{n+1}^{(m)} + \underline{\ell}\, G(\underline{a}_{n+1}^{(m)})$$

and

(3.1.8c) $$\underline{a}_{n+1} = \underline{a}_{n+1}^{(M)} \quad .$$

Initially, all the unknown derivatives of $\underline{a}_0$ are set to zero. Then given $\underline{a}_n$, Gear's method uses equation (3.1.8a) to predict $\underline{a}_{n+1}^{(0)}$. The corrector equation (3.1.8b) is then iterated without changing the Jacobian matrix until the number of iterations M equals 3. If convergence is not obtained, the Jacobian matrix $\partial f/\partial y$ is evaluated again. If this happens a second time, the step is reduced by one-half and the process starts again. An estimate of the local error is obtained by taking the difference of the predicted and corrected values at each step. The stepsize is increased or decreased by a factor of 2 depending on the error tolerence. For changing the order k, the maximum step for orders k-1, k and k+1 is calculated. Whenever the stepsize or the order is changed, k steps must be taken before we can change either the stepsize or the order.

3.2 Fourth Order Exponentially Fitted Method

Liniger and Willoughby [19] introduced the concept of exponential fitting. Recently Bjurel et al. [2] have defined multistep methods which are exponentially fitted in the complex $h\lambda$-plane. Liniger and Willoughby [19] have developed exponentially fitted methods of order three. In this section, we develop a new fourth order exponentially fitted method.

Consider the following integration formula $F^{(4)}$ of order $p = 4$

$$F^{(4)}: y_{n+2} = y_{n+1} + \frac{h}{240}\left(A\dot{y}_n + B\dot{y}_{n+1} + C\dot{y}_{n+2}\right) + \frac{h^2}{240}\left(D\ddot{y}_n + E\ddot{y}_{n+1} + F\ddot{y}_{n+2}\right) + T \tag{3.2.1a}$$

where

$$\begin{aligned} A &= 50a + 26b - 65 , \\ B &= -40a - 52b + 220 , \\ C &= -10a + 26b + 85 , \\ D &= 20a + 13b - 30 , \\ E &= 40a , \\ F &= -13b \end{aligned} \tag{3.2.1b}$$

and a, b are real parameters. The truncation error T is given by

$$T = \frac{(23 - 13b - 10a)}{1440} h^5 y_n^{(5)} + 0(h^6)$$

showing that (3.2.1a) is a fourth order method at least. For a=b=1, formula $F^{(4)}$ reduces to a sixth order strongly stable formula given by

$$y_{n+2} = y_{n+1} + \frac{h}{240} (11\dot{y}_n + 128\dot{y}_{n+1} + 101\dot{y}_{n+2}) + \frac{h^2}{240} (3\ddot{y}_n + 40\ddot{y}_{n+1} - 13\ddot{y}_{n+2}) .$$

For a = 1/2 and b = 20/13, formula $F^{(4)}$ reduces to the well known formula associated with the second diagonal Pade approximation given by

$$y_{n+1} = y_n + \frac{h}{2} (\dot{y}_{n+1} + \dot{y}_n) - \frac{h^2}{12} (\ddot{y}_{n+1} - \ddot{y}_n) .$$

The multistep method $F^{(4)}$, when applied to the scalar equation $\dot{y} = \lambda y$, $Re(\lambda) < 0$, yields the following difference equation

$$(3.2.2a) \qquad y_{n+2}(240 - CQ - FQ^2) + y_{n+1}(-240 - BQ - EQ^2) + y_n(-AQ - DQ^2) = 0$$

where $Q = h\lambda$. Its characteristic polynomial $\chi(z;Q)$ is

$$(3.2.2b) \qquad \chi(z;Q) = z^2(240 - CQ - FQ^2) + z(-240 - BQ - EQ^2) - AQ - DQ^2$$

The polynomial has two roots, both of which must be less than 1 in modulus. The principal root should approximate exp(Q). The most stable choice for the parasatic root is zero. Thus, we choose the parameters in $\chi(z;Q)$ so that

(3.2.3a) $\chi(\exp(Q);Q) = 0$

and

(3.2.3b) $\chi(0;\ Q) = 0$.

Solving (3.2.3) for a and b, we obtain for $Q \neq -2.5$

$$a = (65 + 30Q - 13(2 + Q)S(Q))/(50 + 20Q)$$

and

$$b = S(Q)$$

where

$$S(Q) = N(Q)/D(Q) ,$$

$$N(Q) = (5 + 2Q)(\exp(Q)(85Q - 240) + 220Q + 240) - Q(65 + 30Q)p(Q) ,$$

$$D(Q) = 13Q((5 + 2Q)(\exp(Q)(Q - 2) + 4) - (2 + Q)p(Q))$$

and

$$p(Q) = \exp(Q) + 4(1 - Q) .$$

Since a and b are chosen to make the parasatic root zero, the principal root of (3.2.2b) is

$$r(Q) = \frac{240 + BQ + EQ^2}{240 - CQ - FQ^2} \quad .$$

To prove that $F^{(4)}$ is A-stable, we use the following theorem due to Liniger and Willoughby [20]:

Theorem

Let $r(Q)$ be a non-constant analytic function for Real $Q < 0$. Then $|r(Q)| < 1$ for Real $Q < 0$ if and only if both of the following are true:

(3.2.4a) $\quad |r(Q)| \leq 1 \quad$ on $\quad$ Real $Q = 0$

and

(3.2.4b) $\quad \lim_{|Q| \to \infty} |r(Q)| \leq 1 \quad .$

Theorem

The necessary and sufficient conditions for the formula $F^{(4)}$ to be A-stable are

(3.2.5a) $\quad |a/b| \leq 13/40 \quad ,$

(3.2.5b) $\quad |C| \leq |B| \quad ,$

(3.2.5c) $\quad 0 \leq C$

and

(3.2.5d) $\quad 0 \leq b$

where B and C are defined by (3.2.1b).

Proof

$F^{(4)}$ is A-stable if and only if $|r(Q)| < 1$ for Real $Q < 0$. First we show that if $|r(Q)| < 1$, then conditions (3.2.5) hold.

Let $Q \to -\infty$ on the real axis. Then we must have $|E| \leq |F|$, which using (3.2.1b) yields equation (3.2.5a).

Since $|r(Q)| < 1$ for Real $Q < 0$, all the poles of $r(Q)$ are in the right half complex plane. Thus the zeros z_j of $240 - CQ - FQ^2$, satisfy Real $z_j \geq 0$, which is true if and only if $F \leq 0$ and $C \geq 0$. Hence (3.2.5c) and (3.2.5d) hold.

Finally, $|r(Q)| < 1$ implies $|B + EQ| \leq |C + FQ|$. Letting $Q \to 0$ gives conditions (3.2.5b).

To show that these conditions imply that $|r(Q)| < 1$ for Real $Q < 0$, note that conditions (3.2.5c) and (3.2.5d) imply that $r(Q)$ is analytic in the left complex plane. We finally note that conditions (3.2.5a) and (3.2.5b) insure that the conditions of the previous theorem are satisfied.

Q.E.D.

CHAPTER IV

TEST PROBLEMS AND CONCLUSIONS

4.1 Introduction

In this chapter, results of test problems, using both the Liniger and Willoughby method and the new fourth order method are given. The test problems chosen represent both linear and non-linear systems, with moderate and highly stiff combinations. Since $|h\partial f/\partial y|$ is large for stiff systems, Newton's method is used for solving the implicit difference equations, which arise in applying each method. The inputs to each computer program were the initial values, eigenvalues at $x = 0$, the error tolerance and the Jacobian $\partial f/\partial y$. For systems whose Jacobian is a function of time, we used the eigenvalues found for $x = 0$. The first solution point for the fourth order method was computed by formula $F^{(3)}$, using the stepsize $h/8$, where h is the stepsize for the fourth order method.

4.2 Computational Equations for the Liniger and Willoughby Method

Applying formula $F^{(1)}$ to (2.1.1), the following difference equation is obtained

$$y_{n+1} - (1-\mu)hf_{n+1} = y_n + \mu hf_n \ . \tag{4.2.1}$$

Equation (4.2.1) is solved by the following Newton iteration

$$(I-(1-\mu)hJ_{n+1}^{(p)})\Delta_{n+1}^{(p)} = y_n + \mu hf_n + (1-\mu)hf_{n+1}^{(p)} - y_{n+1}^{(p)} \tag{4.2.2}$$

where

$$\Delta_{n+1}^{(p)} = y_{n+1}^{(p+1)} - y_n^{(p)} \ , \quad p=0,1,2,\ldots,M \ ,$$

$$J_{n+1}^{(p)} = \partial f(x_{n+1}, y_{n+1}^{(p)})/\partial y \ ,$$

$$f_{n+1}^{(p)} = f(x_{n+1}, y_{n+1}^{(p)})$$

with $y_{n+1}^{(0)} = y_n$ and $y_n = y_n^{(M)}$. The number M, which is the maximum number of iterations, is chosen to be p+1, where

$$\left| y_{n+1}^{(p+1)} - y_{n+1}^{(p)} \right| < \varepsilon$$

with ε a fixed tolerence.

Applying formula $F^{(2)}$ to (2.1.1), the following difference equation is obtained

$$y_{n+1} - h(1+a)f_{n+1}/2 + h^2(b+a)J_{n+1}f_{n+1}/4$$

$$- y_n - h(1-a)f_n/2 - h^2(b-a)J_nf_n/4 = 0 \ .$$

By using the Newton iteration, we obtain

$$(4.2.3) \qquad (4I - 2h(1+a)J_{n+1}^{(p)} + h^2(b+a)[J_{n+1}^{(p)}]^2)\Delta_{n+1}^{(p)}$$

$$= 4(y_n - y_{n+1}^{(p)}) + h^2(2(1+a)I - h(b+a)J_{n+1}^{(p)})f_{n+1}^{(p)}$$

$$+ 2h(1-a)f_n + h^2(b-a)J_nf_n \ .$$

The computational equation for $F^{(3)}$ follows from (4.2.3) by substituting $b = 1/3$ and $c = a$.

4.3 The Computational Equation for the Fourth Order Exponentially Fitted Method

Applying formula $F^{(4)}$ (3.2.1a) to (2.1.1), the following difference equation is obtained

$$y_{n+2} = y_{n+1} + h(Af_n + Bf_{n+1} + Cf_{n+2})/240$$
$$+ h^2(DJ_nf_n + EJ_{n+1}f_{n+1} + FJ_{n+2}f_{n+2})/240$$

where the coefficients A, B, C, D, E and F are defined by (3.2.1b).

Following Liniger and Willoughby's technique of linearizing the approximations (see section 4.2), the computational equation becomes

$$(240I - hCJ_{n+2}^{(p)} - h^2F[J_{n+2}^{(p)}]^2)\Delta_{n+2}^{(p)} = 240(y_{n+1} - y_{n+2}^{(p)})$$
$$+ h(AI + hDJ_n)f_n + h(BI + hEJ_{n+1})f_{n+1}$$
$$+ h(CI + hFJ_{n+2}^{(p)})f_{n+2}^{(p)}$$

where $y_{n+2}^{(0)} = y_{n+1}$ is used for starting each iteration.

4.4 Ehle's Linear Problem I [8]

(4.4.1) $\dot{y} = -y + 95z$, $y(0) = 1$,

$\dot{z} = -y - 95z$, $z(0) = 1$, $x \in [0,1]$.

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -96$, $\lambda_1 = -2$.

True solution: $y = (95 \exp(-2x) - 48 \exp(-96x))/47$

$z = (48 \exp(-96x) - \exp(-2x))/47$.

Table 2

The Solutions of Equation (4.4.1)

STEP	METHOD	y(1) ($\lvert$ERROR$\rvert$)	$z(1) \times 10^2$ ($\lvert$ERROR$\rvert \times 10^2$)
0.0625	$F^{(3)}$	0.2735430 (7×10^{-6})	-0.2879400 (7×10^{-6})
	$F^{(4)}$	0.2735503 (3×10^{-7})	-0.2879477 (4×10^{-7})
0.03125	$F^{(3)}$	0.27354952 (5×10^{-7})	-0.28794687 (5×10^{-7})
	$F^{(4)}$	0.27355005 (1×10^{-8})	-0.28794742 (1×10^{-8})
0.015625	$F^{(3)}$	0.27355000 (4×10^{-8})	-0.28794737 (1×10^{-8})
TRUE SOLUTION		0.27355004	$-0.2879474 1\times10^{-2}$

4.5 Ehle's Linear Problem II [8]

(4.5.1) $\dot{y} = -y + 95z , \quad y(0) = 1 ,$

$\dot{z} = -y - 95z , \quad z(0) = -1/95 , \quad x \in [0,1].$

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -96$, $\lambda_2 = -2$.

True solution: $y = \exp(-2x)$

$z = -(\exp(-2x))/95 .$

Table 3

The Solutions of Equation (4.5.1)

STEP	METHOD	$y(1)$ ($\lvert$ERROR$\rvert$)	$z(1) \times 10^2$ ($\lvert$ERROR$\rvert \times 10^2$)
0.0625	$F^{(3)}$	0.135331816 (3×10^{-6})	-0.14245454 (3×10^{-6})
	$F^{(4)}$	0.1353354305 (1×10^{-7})	-0.1424583479 (1×10^{-7})
0.03125	$F^{(3)}$	0.1353350304 (2×10^{-7})	-0.14245792 (2×10^{-7})
	$F^{(4)}$	0.1353352887 (5×10^{-9})	-0.1424581987 (5×10^{-9})
0.015625	$F^{(3)}$	0.1353352666 (1×10^{-8})	-0.14245817 (2×10^{-8})
TRUE SOLUTION		0.1353352832	$-0.1424581928\times10^{-2}$

4.6 Fowler and Warten's Linear Problem [9]

(4.6.1) $\dot{y} = -2000y + 1000z + 1 , \quad y(0) = 0 ,$

$\dot{z} = y - z , \quad z(0) = 0 , \quad x \in [0,4].$

Eigenvalues of the Jacobian at x = 0 : $\lambda_1 = -2000.5$, $\lambda_2 = -0.5$.

Table 4

The Solutions of Equation (4.6.1)

STEP	METHOD	$y(4) \times 10^3$ ($\lvert ERROR \rvert \times 10^3$)	$z(4) \times 10^3$ ($\lvert ERROR \rvert \times 10^3$)
0.0625	$F^{(3)}$	0.93226472 (6×10^{-8})	0.864563300 (1×10^{-7})
	$F^{(4)}$	0.93226466 (EXACT)	0.8645631887 (1×10^{-9})
0.03125	$F^{(3)}$	0.93226467 (1×10^{-8})	0.864563203 (1×10^{-8})
	$F^{(4)}$	0.93226466 (EXACT)	0.8645631898 (1×10^{-10})
0.015625	$F^{(3)}$	0.93226466 (EXACT)	0.864563191 (2×10^{-9})
TRUE SOLUTION R-K 4 RALSTON h = 0.001		0.93226466×10^{-3}	$0.8645631899\times10^{-3}$

4.7 Variable Eigenvalue Linear Problem [10]

$$\dot{Y} = AY \,, \quad Y(0) = [2,0]^t \,, \quad x \in [0,1] \tag{4.7.1}$$

where

$$A = -\frac{1}{2}\begin{pmatrix} 1+c & 1-c \\ 1-c & 1+c \end{pmatrix} .$$

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -c$, $\lambda_2 = -1$.

True solution: $y_1 = \exp(-x) + \exp(-cx)$

$y_2 = \exp(-x) - \exp(-cx)$.

We consider the particular case $c = 16$.

Table 5

The Solutions of Equation (4.7.1)

STEP	METHOD	$y_1(1)$ ($\lvert$ERROR$\rvert$)	$y_2(1)$ ($\lvert$ERROR$\rvert$)
0.0625	$F^{(3)}$	0.367879437 (1×10^{-7})	0.36787921266 (1×10^{-7})
	$F^{(4)}$	0.3678795562 (2×10^{-9})	0.36787933114 (2×10^{-9})
0.03125	$F^{(3)}$	0.36787954641 (7×10^{-9})	0.36787932134 (7×10^{-9})
	$F^{(4)}$	0.36787955378 (8×10^{-11})	0.3678793287 (1×10^{-10})
TRUE SOLUTION		0.36787955370	0.3678793286

4.8 Lapidus's Linear Problem [15]

(4.8.1) $\dot{Y} = AY$, $Y(0) = [2,1,2]^t$, $x \in [0,1]$

where

$$A = \begin{bmatrix} -0.1 & -49.9 & 0 \\ 0 & -50 & 0 \\ 0 & 70 & -120 \end{bmatrix}$$

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -120$, $\lambda_2 = -50$, $\lambda_3 = -0.1$.

True solution: $y_1 = \exp(-0.1x) + \exp(-50x)$

$y_2 = \exp(-50x)$

$y_3 = \exp(-50x) + \exp(-120x)$.

Table 6

The Solutions of Equation (4.8.1)

STEP	METHOD	$y_1(1)$ (|ERROR|)	$y_k(1)$ k=2,3 (|ERROR|)
0.062[illegible]	$F^{(3)}$	0.904837417863 (2×10^{-10})	0.14×10^{-26} (0.19×10^{-21})
	$F^{(4)}$	0.90483741803630 (4×10^{-13})	0.25×10^{-17} (0.25×10^{-17})
0.031[illegible]5	$F^{(3)}$	0.904837418022840 (1×10^{-11})	0.11×10^{-21} (0.8×10^{-22})
	$F^{(4)}$	0.9048374180359725 (1×10^{-14})	0.31×10^{-21} (0.1×10^{-22})
TRUE SOLUTION		0.9048374180359595	0.19×10^{-21}

4.9 Fowler and Warten's Non-linear Problem [9]

$$\dot{y} = -10\dot{z} + 3000\,(1-y^2), \qquad y(0) = 1, \tag{4.9.1}$$

$$\dot{z} = 0.04(1-z) - (1-y)z + 10^{-4}(1-y)^2, \quad z(0) = 1, \quad x \in [0, 2.6]$$

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -6000$, $\lambda_2 = -0.04$.

Table 7

The Solutions of Equation (4.9.1)

STEP	METHOD	y(2.6) |(ERROR)|	z(2.6) × 10 (|ERROR|×10)
0.1625	$F^{(3)}$	0.999939925473 (EXACT)	0.9876685530 (4×10^{-9})
	$F^{(4)}$	0.999939925473 (EXACT)	0.98766854924 (1×10^{-11})
0.08125	$F^{(3)}$	0.999939925473 (EXACT)	0.9876685497 (5×10^{-10})
	$F^{(4)}$	0.999939925473 (EXACT)	0.98766854924 (1×10^{-11})
TRUE SOLUTION R-K 4 RALSTON h = 0.00040625		0.999939925473	$0.98766854923\times10^{-1}$

4.10 Lawson's Non-linear Problem [16]

(4.10.1) $$\dot{y} = (-1+z^2)y + (1+z)z \ , \quad y(0) = -1$$

$$\dot{z} = -y + (-19+2y+z^2)z \ , \quad z(0) = 1 \ , \quad x \in [0,1].$$

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -19.9$, $\lambda_2 = -0.1$.

Table 8

The Solutions of Equation (4.10.1)

STEP	METHOD	y(1) (|ERROR|)	z(1) × 10 (|ERROR|×10)
0.03125	$F^{(3)}$	-0.33062729 (4×10^{-6})	0.17849359 (2×10^{-6})
	$F^{(4)}$	-0.33063158 (7×10^{-7})	0.17849585 (3×10^{-7})
0.015625	$F^{(3)}$	-0.33063060 (3×10^{-7})	0.178495338 (2×10^{-7})
	$F^{(4)}$	-0.33063088 (3×10^{-8})	0.17849548 (1×10^{-7})
TRUE SOLUTION R-K 5 LAWSON [16]		-0.33063085	0.1784955×10^{-1}

4.11 Gear's Non-linear Problem [11]

$$\dot{y} = -(55 + z)y + 65z \,, \quad y(0) = 1 \,,$$

$$\dot{z} = 0.0785\,(y-z) \,, \quad z(0) = 1 \,,$$

$$\dot{u} = 0.1\,y \,, \quad u(0) = 0 \,, \quad x \in [0,1]. \tag{4.11.1}$$

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -55$,

$\lambda_2, \lambda_3 = 0.0625 \pm 0.01\,i$.

Table 9

The Solutions of Equation (4.11.1)

STEP	METHOD	y(1) (|ERROR|)	z(1) (|ERROR|)	u(1) (|ERROR|)
0.125	$F^{(3)}$	1.1955191535 (8×10^{-9})	1.013991515 (7×10^{-9})	0.118510694 (1×10^{-8})
	$F^{(4)}$	1.19551914432 (9×10^{-10})	1.01399150753 (8×10^{-10})	0.1185106830 (1×10^{-9})
0.0625	$F^{(3)}$	1.19551914618 (1×10^{-9})	1.0139915091 (8×10^{-10})	0.1185106852 (1×10^{-9})
	$F^{(4)}$	1.19551914499 (2×10^{-10})	1.01399150814 (2×10^{-10})	0.1185106838 (2×10^{-10})
TRUE SOLUTION GEAR'S PROGRAM [12]		1.195519145189	1.01399150830	0.1185106840

4.12 Liniger and Willoughby's Non-linear Problem [19]

(4.12.1) $$\dot{y} = a_1 a_5 - (a_1 + a_2(y+a_4)(y+1))p(y,z) \ , \ y(0) = 0$$

$$\dot{z} = a_5 - (1 + a_3 z^2)p(y,z) \ , \ z(0) = 0, \ x \in [0,100].$$

where

$$p(y,z) = a_5 + y + z \ ,$$

$$a_1 = a_5 = a_3 = 1 \ , \ a_4 = 1000 \ \text{ and } \ a_5 = 0.01 \ .$$

Eigenvalues of the Jacobian at $x = 0$: $\lambda_1 = -1012$, $\lambda_2 = -0.01$.

Table 10

The Solutions of Equation (4.12.1)

STEP	METHOD	y(100) |Difference between F^4(h=0.0625) and y_n|	z(100) |ERROR|
0.125	$F^{(3)}$	-0.99163973 (2×10^{-6})	0.9833336 (3×10^{-6})
	$F^{(4)}$	-0.991641966 (8×10^{-8})	0.983336240 (1×10^{-7})
0.0625	$F^{(3)}$	-0.991640918 (1×10^{-6})	0.983335044 (1×10^{-6})
	$F^{(4)}$	-0.991642044 (ASSUMED EXACT)	0.983336329 (3×10^{-8})
TRUE SOLUTION R-K 4 ODEN [21]		NOT REPORTED	0.983336361

4.13 Moore's System [7]

$$
\begin{aligned}
(4.13.1)\qquad \dot{y} &= -100z + 100v - c(z-v)^2 , & y(0) &= 1 ,\\
\dot{z} &= y , & z(0) &= 0 ,\\
\dot{u} &= (100z - 101v + c(z-v)^2 - dv^2)/100 , & u(0) &= 0 ,\\
\dot{v} &= u , & v(0) &= 0 , \quad x \in [0,65].
\end{aligned}
$$

where $c = d = 0$ for the linear case.

Eigenvalues of the Jacobian at $x = 0$: $\lambda_{1,2} = -0.5 \pm 10i$

$\lambda_{3,4} = -0.005 \pm 0.1i$.

Table 11

The Solutions of Equations (4.13.1)

$c = d = 0$

STEP	METHOD	$y(65)\times10^{2}$	$z(65)\times10^{1}$	$u(65)\times10^{2}$	$v(65)\times10$ ($\lvert ERROR \rvert$)
0.0625	$F^{(3)}$	0.70	0.12	0.70	0.126377 (7×10^{-1})
	$F^{(4)}$	0.70	0.12	0.70	0.126377 (7×10^{-1})
TRUE SOLUTION R-K 4 ODEN [21]		NOT REPORTED	NOT REPORTED	NOT REPORTED	0.196403

4.14 Conclusions

We have discussed those methods, which we regard as being the most important for solving stiff ordinary differential equations. There are many other methods in the literature [Bjurel et al. [2]), but these appear to be too special in the sense, that they are applicable only to problems satisfying very special constraints. In fact, even those methods surveyed in this thesis are not applicable in general.

Of the single step methods, the modified R-K methods simply increase the stability boundary. Calahan's method and Lawson's method for linear systems appear to be most promising among R-K type methods. Dahlquist's smoothing technique is quite effective for damping the oscillations in linear systems with constant coefficients. Experimental results reported by Lapidus et al. [15] indicate that the Liniger and Willoughby method is the best among single step methods, although Calahan's method is computationally faster for moderately stiff systems. Also, it was found that for Moore's system (4.13), which has large eigenvalues near the imaginary axis, the Liniger and Willoughby method's performance was quite poor.

The most promising multistep methods seem to be Gear's method and Dahlquist's SAPS method [7]. Oden [21] reports that for the SAPS method, the number of function evaluations are less than that of the Gear's method. We

note that Oden has chosen only those stiff systems, which satisfy conditions imposed by the SAPS method. Furthermore, Gear's method is applicable to both stiff and non-stiff systems and it has both automatic stepsize and variable order control. Thus among multistep methods, we would prefer to use Gear's method.

The second objective of this thesis was to experiment with the Liniger and Willoughby methods. In the process of experimentation, we developed a new fourth order method. Since the new method involves extra function evaluations, the cost of n steps for both the $F^{(3)}$ and $F^{(4)}$ methods, is given in Table 12. Here M represents the maximum number of iterations per step and ℓ represents the dimension of the system. We assume that the overhead is the same for both methods.

Table 12

The Cost of n Steps for $F^{(3)}$ and $F^{(4)}$

TYPE OF STIFF SYSTEM	METHOD	MATRIX INVERSION	MATRIX. SCALAR	MATRIX. VECTOR	MATRIX-ADDITION	VECTOR-ADDITION
NON-LINEAR $\dot{y} = f(y)$	$F^{(4)}$	$nM\ell^3$	$9nM\ell^2$	$3nM\ell^2$	$9nM\ell^2$	$6nM\ell$
	$F^{(3)}$	$nM\ell^3$	$6nM\ell^2$	$2nM\ell^2$	$5nM\ell^2$	$6nM\ell$
LINEAR WITH CONSTANT COEFFICIENTS	$F^{(4)}$	ℓ^3	ℓ^2	$n\ell^2$	0	$8n\ell$
	$F^{(3)}$	ℓ^3	ℓ^2	$n\ell^2$	0	$5n\ell$

For the non-linear systems, we see that the cost of $F^{(4)}$ is higher in the columns of MATRIX.SCALAR and MATRIX-ADDITION. For linear systems with constant coefficients, we note that the cost of n steps for $F^{(4)}$ is more in the VECTOR-ADDITION column only. In fact, from Tables 2-6, we obtain that $F^{(4)}$'s performance with a stepsize 2h is the same as the performance of $F^{(3)}$ with a stepsize h. To check the tradeoff of high cost of $F^{(4)}$ for non-linear systems, the execution times of both methods, for the non-linear highly stiff system (4.12) are given in Table 13. Here error represents the maximum error in either y(100) or z(100).

Table 13

The Excess Time of $F^{(4)}$ Over $F^{(3)}$

STEP	METHOD	|ERROR|	EXECUTION TIME IN SECS.	EXCESS TIME
0.125	$F^{(3)}$	3×10^{-6}	28.86	
	$F^{(4)}$	1×10^{-7}	31.26	2.40 secs.
0.0625	$F^{(3)}$	1×10^{-6}	64.36	
	$F^{(4)}$	3×10^{-8}	66.11	1.75 secs.

Thus we see that even with $h = 0.0625$, $F^{(3)}$ does not yield the accuracy as is attained by $F^{(4)}$ with $h = 0.125$. We note that the increase in execution time for $h = 0.0625$ is just 2.7%, while the error is reduced by two digits at least.

In conclusion, noting the high cost and higher accuracy of $F^{(4)}$, we would say that $F^{(4)}$ should be used for linear systems with constant coefficients. For non-linear systems, $F^{(4)}$ should only be used, if high accuracy is desired.

BIBLIOGRAPHY

1. Ahlfors, L.V. [1953], Complex Analysis, McGraw Hill, N.Y.

2. Bjurel, G. et al. [1970], "Survey of Stiff Ordinary Differential Equations", Report NA 70.11, Royal Institute of Technology, Stockholm, Sweden.

3. Bjurel, G. [1969], "Modified Linear Multistep Methods for a Class of Stiff Ordinary Differential Equations", Report NA 69.02, Royal Institute of Technology, Stockholm, Sweden.

4. Calahan, D.A. [1967], "A Stable, Accurate Method of Numerical Integration for Non-linear Systems", Proc. IEEE, Vol. 55, pp.2016-2017.

5. Curtiss, C.F. and Hirschfelder, J.O. [1952], "Integration of Stiff Equations", Proc. Natl. Acad. Sci. U.S., Vol.38, pp.235-243.

6. Dahlquist, G. [1963], "A Special Stability Problem for Linear Multistep Methods", BIT, Vol.3, pp.27-43.

7. ————————. [1968], "A Numerical Method for Some Ordinary Differential Equations with Large Lipschitz Constants", Proc. IFIP Congress, pp.183-186, Edinburgh, Scotland.

8. Ehle, B.L. [1969], "On Padé Approximations to the Exponential Function and A-stable Methods for the Numerical Solution of Initial Value Problems", Dept. of AACS, University of Waterloo, Res. Rep., CSRR 2010.

9. Fowler, M.E. and Warten R.M. [1967], "A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues", IBM J. Res. Develop., Vol.11, pp.537-543.

10. Gable, G.F. [1968], "A Predictor-Corrector Method using Divided Differences", Technical Report No.5, Dept. of Computer Sc., Univ. of Toronto.

11. Gear, C.W. [1968], "The Automatic Integration of Stiff Ordinary Differential Equations", Proc. IFIP Congress 68, pp.81-85, Edinburgh, Scotland.

12. ————————. [1971], "Numerical Initial Value Problems in Ordinary Differential Equations", Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

13. ————————. [1967], "The Numerical Integration of Ordinary Differential Equations", Math. Comp, 21 pp.146-156.

14. Jain, M.K. and Srivastava, V.K. [1970], "High Order Stiffly Stable Methods for Ordinary Differential Equations", Report No.394, Urbana, Ill., Dept. of Computer Sci., Univ. of Illinois.

15. Lapidus, L. and Seinfeld, J.H. [1971], "Numerical Solution of Ordinary Differential Equations", Academic Press, N.Y.

16. Lawson, J.D. [1966], "Generalized Runge-Kutta Processes for Stable Systems with Large Lipschitz Constants", SIAM. J. Num. Anal., Vol.3, No.4, pp.593-597.

17. Lindberg, B. [1969], "On Smoothing and Extrapolation for the Trapezoidal Rule", Res. Report, The Royal Institute of Technology, Stockholm, Sweden.

18. ————————. [1971], "On Smoothing for the Trapezoidal Rule", Report NA 71.31, The Royal Institute of Technology, Stockholm, Sweden.

19. Liniger, W. and Willoughby, R.A. [1967], "Efficient Numerical Integration of Stiff Systems of Ordinary Differential Equations", Res. Report RC 1970, IBM, N.Y.

20. Liniger, W. [1969], "Global Accuracy and A-stability of One and Two Step Integration Formulae for Stiff Ordinary Differential Equations", Conference on the Numerical Solution of Differential Equations, Springer - Verlag, Berlin, pp.188-193.

21. Oden, L. [1971], "An Experimental and Theoretical Analysis of the SAPS - method for Stiff Ordinary Differential Equations", Report NA 71.28, Royal Institute of Technology, Stockholm, Sweden.

22. Rosenbrock, H.H. [1963], "Some General Implicit Processes for the Numerical Solution of Differential Equations", Computer Journal, Vol.5, pp.329-330.